COMS E6121 Reliable Software  
Petre Lukarov  
[pbl2108@columbia.edu](mailto:pbl2108@columbia.edu)

Final Project Proposal

**Problem Statement:**

Today’s web-applications rely heavily on AJAX, meaning only certain parts are updated and not the entire page. During this partial updates the DOM tree is not destroyed, but certain nodes are updated. JavaScript is used to update the DOM tree. These updates however can introduce new HTML that is invalid, or remove nodes without detaching event handlers. I propose a system that checks the modified part of the DOM tree, and reports problems to the developer (user).   
The initial system shall support:  
- Ability to add own checkers according to the DOM tree changes  
- Detection of removing DOM elements that have not their handlers detached  
- Invalid HTML, i.e. not compliant to the current document type used

**Approach:**

A JavaScript based system can track the changes of the DOM tree and determine what’s new in the tree. Once the modified nodes are identified, the changes are checked to see if they cause a problem. The problem could be invalid HTML as well as the event handlers are not detached before removing the elements from the DOM tree.

The system can be provided as extension or plugin to the browser. Almost all modern browsers support plugins.

**Novelty Claim:**

At the moment there exists a tool for checking the validity of an HTML page, however it only checks the (static) content that is received from the server. This is the online W3C validator: <http://validator.w3.org/>.

Also there is a memory leaking tool that will catch memory leaks resulting from deleting a DOM element that has event handlers not detached. Here is the link: <http://home.wanadoo.nl/jsrosman/>. However this system reports memory leaks and does not give the reason for the leak. The developer has to figure out by herself what’s the reason, which might be difficult.

**Timeline:**

1. The system that traces the changes in the DOM will be implemented first. (2-3 weeks)
2. The checker for deleting DOM elements without detaching handlers will be implemented (2 weeks)
3. The checker for checking invalid HTML will be implemented (2 weeks)

**Questions from the professor:**

1. How're you going to evaluate your tool?
2. What websites/web apps will you use?
3. How do you define success for your project?

**My answers:**

1.

A defined set of web-pages that we are known to append invalid HTML, or do not detach all handlers, can be used to evaluate the correctness of the tool. Moreover the invalid content can be divided into generic groups of invalid content. For example one type would be adding "block" HTML elements (such as <div>) as children to "inline" HTML elements (such as <span>). Invoking ***addHTML***should cause a <div> to be added to a <span> and the tool should report this as a problem.

*<span id="span1">*

*</span>*

*function addHTML()*

*{*

*var span1 = $get("span1");*

*span1.innerHTML = "<div id='div1'>Sample Content</div>";*

*}*

*The following example shows removing HTML without detaching the handlers first. The function "addBtnHadnler" is called first to attach the "clickHandler" to the button's click event. Now if the "removeButton" function is called before the handler is detached, the tool should catch this problem and report it to the user.*

*<div id="divContainer1">*

*<button id="btn1">Click Me</button>*

*</div>*

*function addBtnHandler()*

*{*

*var btn1 = $get("btn1");*

*$addHandler(btn1, "click", clickHandler);*

*}*

*function clickHandler()*

*{*

*alert("Button Clicked");*

*}*

*function removeButton()*

*{*

*var containerDiv = $get("divContainer1");*

*containerDiv.innerHTML = "";*

*}*

2. While working at my previous company we created reusable ASP.NET components that programmatically a lot of HTML and appended it to the DOM tree. This HTML was not always compliant, even though it worked correctly - browsers are very forgiving. Moreover end-users (other developers) can use custom content which can also be invalid. Here is a link to one of the demo web-pages showing a *RadRotator* control switching user-defined content.

<http://demos.telerik.com/aspnet-ajax/rotator/examples/overview/defaultcs.aspx>

Each web-page can be tested for validity.

Many company web-sites extensively load dynamic content ([www.apple.com](http://www.apple.com), [www.facebook.com](http://www.facebook.com), [www.yahoo.com](http://www.yahoo.com), [www.google.com](http://www.google.com) and many more), so they can be tested also.

3.

The tool is successful if it catches the occurrences of invalid HTML inserted to the tree dynamically. Also the tool must successfully catch all changes in the DOM and try to validate each change. The conditions on which the changes are validated should be changeable and a user can define her own policy for validation (that is the user should be able to add custom validation logic).